BAB 2

TINJAUAN PUSTAKA

2.1 Tinjauan Perusahaan

2.1.1 Sejarah Badan Perpustakaan dan Kearsipan Daerah

Secara historis, lembaga perpustakaan daerah ditingkat Provinsi Jawa Barat yang saat ini menjadi Badan Perpustakaan dan Kearsipan Daerah Provinsi Jawa Barat, sebelumnya mengalami beberapa kali perubahan nama.

Sebagai cikal bakalnya bernama Perpustakaan Negara yang berdiri pada tanggal 23 Mei 1956, dibentuk berdasarkan Surat Keputusan Mentri Pendidikan Pengajaran dan Kebudayaan nomor 29103/s di 19 provinsi, salah satunya yaitu Bandung yang berlokasi di Jalan Diponegoro serta induk organisasinya adalah Biro Perpustakaan dan Pembinaan Buku. Setelah terbit Surat Keputusan Menteri Pendidikan dan Kebudayaan nomor 095/1967 tanggal 6 Desember 1967, ditetapkan bahwa lembaga perpustakaan merupakan induk organisasi perpustakaan negara, kemudian berdasarkan Surat Keputusan Menteri Pendidikan dan Kebudayaan nomor 079/1975 induk organisasi perpustakaan negara menjadi Pusat Pembinaan Perpustakaan.

Empat tahun kemudian, tepatnya tanggal 29 Mei 1979 Menteri Pendidikan dan Kebudayaan mengeluarakan Surat Keputusan nomor 095/1979 tentang Penetapan Pengalihan Nama Perpustakaan Negara Menjadi Perpustakaan Wilayah, sementara induk organisasinya masih Pusat Pembinaan Perpustakaan.

Adanya penggabungan Pusat Pembinaan Perpustakaan dengan Perpustakaan Nasional Republik Indonesia berdasarkan keputusan presiden nomor 11 tahun 1989 tentang Perpustakaan Nasional RI, pasal 14(1) nama Perpustakaan Wilayah yang ada di setiap provinsi berubah lagi menjadi Perpustakaan Daerah Jawa Barat dan induk organisasinya adalah Perpustakaan Nasional Republik Indonesia yang merupakan Lembaga Pemerintah Non Departemen (LPND).

Setelah terbitnya keputusan presiden nomor 50 tahun 1997 tentang Perpustakaan Nasional Republik Indonesia, maka selaras pasal 16(1) nama perpustakaan daerah Jawa Barat berubah lagi menjadi Perpustakaan Nasional Provinsi Jawa Barat, sedangkan induk organsasinya masih Perpustakaan Nasional Republik Indonesia.

Dengan berlakunya Undang-undang nomor 22 Tahun 1999, pada tahun 2001 Perpustakaan Nasional Provinsi Jawa Barat yang pada awalnya merupakan instansi vertikal perpustakaan nasional Republik Indonesia yang berada di ibukota provinsi dilimpahkan kepada pemerintah Provinsi Jawa Barat.

Kemudian tanggal 12 April 2002, berdasarkan peraturan daerah nomor 6 dibentuk Badan Perpustakaan Daerah Jawa Barat sebagai salah satu lembaga teknis daerah Pemerintah Provinsi Jawa Barat dalam bidang perpustakaan.

Selanjutnya Badan Perpustakaan Daerah Provinsi Jawa Barat yang semula berdiri sendiri sebagai lembaga teknis daerah pemerintah Provinsi Jawa Barat bidang perpustakaan mulai 19 November 2008 mengalami perubahan lagi menjadi Badan Perpustakaan dan Kearsipan Daerah berdasarkan peraturan daerah Provinsi Jawa Barat nomor 22 tahun 2008 tentang organisasi dan tata kerja inspektorat, Badan Perencanaan Pembangunan Daerah, lembaga teknis daerah dan satuan polisi pamong praja Provinsi Jawa Barat (lembaran daerah tahun 2008 nomor 21 seri D, tambahan lembaran daerah nomor 56).

2.1.2 Profil Perpustakaan

Adapun profil dari Badan Perpustakaan dan Kearsipan Daerah adalah sebagai berikut :

1. Nama perpustakaan : Badan Perpustakaan dan Kearsipan Daerah
2. Alamat perpustakaan : Jln. Soekarno Hatta no. 629 Bandung
3. No telp : 022 7310435
4. No fax : 022 7301408
5. Website : bapusipda.jabarprov.go.id

2.1.3 Visi Misi dan Tujuan Perpustakaan

A. Visi Perpustakaan

Perpustakaan dan kerasipan sebagai sumber informasi yang handal menuju masyarakat jawa barat cerdas.

B. Misi Perpustakaan

1. Meningkatkan pembinaan dan pengembangan lembaga perpustakaan dan kearsipan
2. Meningkatkan profesionalisme dan kompetensi SDM pengelola perpustakaan dan kearsipan.
3. Mengembangkan budaya baca masyarakat guna mewujudkan masyarakat belajar (learning society).
4. Mengembangkan budaya sadar arsip.
5. Mengembangkan, mengelola, dan melestarikan bahan perpustakaan dan arsip sebagai khasanah informasi dan pengetahuan.
6. Menyelenggarakan layanan perpustakaan dan kearsipan berbasis teknologi informasi dan komunikasi.

C. Tujuan Perpustakaan

Memberikan layanan kepada pemustaka, meningkatkan pemberdayaan dan kegemaran membaca, serta memperluas wawasan dan pengetahuan untuk mencerdaskan kehidupan bangsa.

2.1.4 Tugas Pokok dan Fungsi

Selaras dengan peraturan Gubernur Jawa Barat nomor 57 tahun 2009 Badan Perpustakaan dan Kearsipan Daerah Provinsi Jawa Barat sebagai lembaga teknis daerah pemerintahan Provinsi Jawa Barat mempunyai tugas

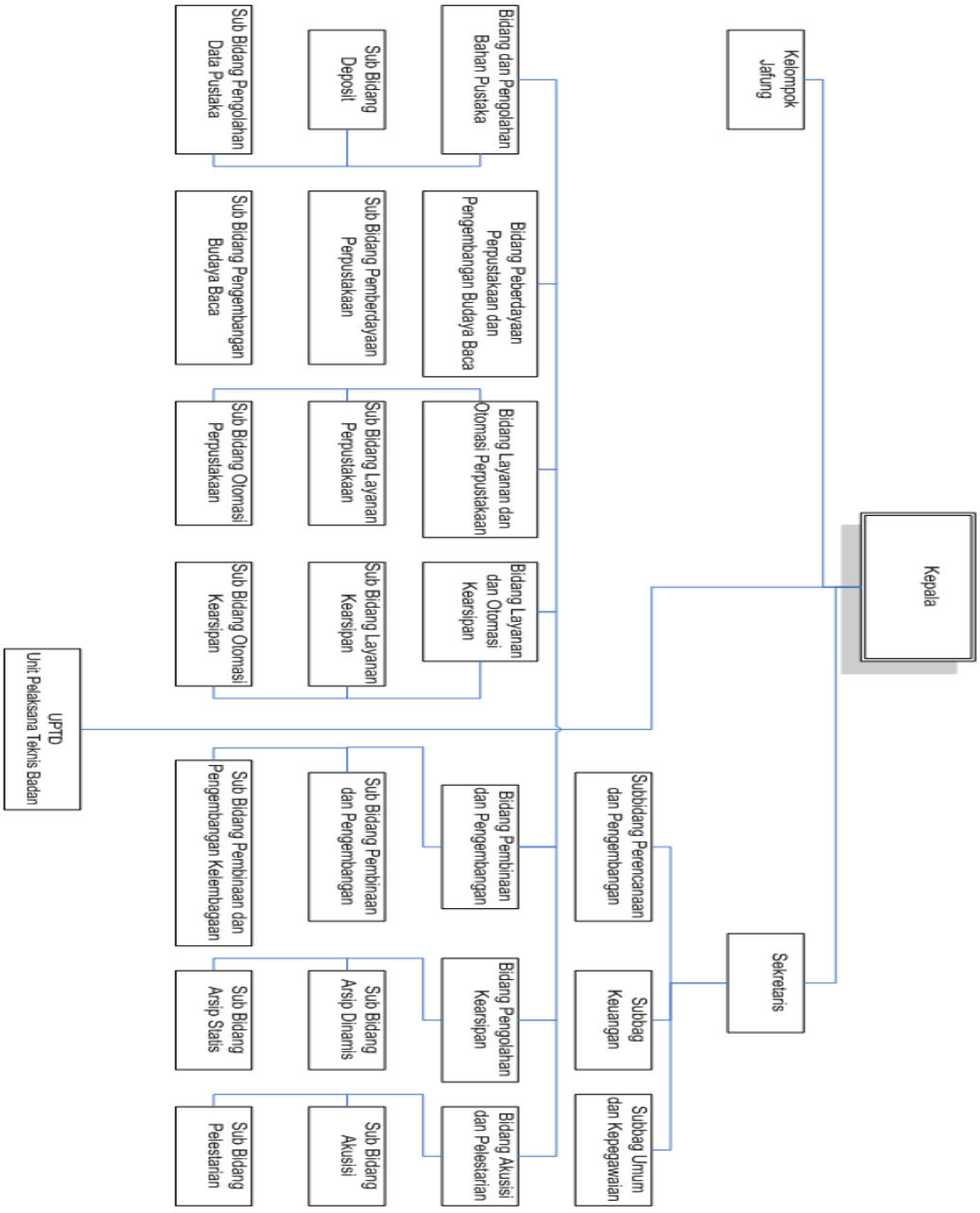
pokok melaksanakan perumusan dan pelaksanaan kebijakan daerah bidang perpustakaan dan kearsipan daerah berdasarkan asas otonomi, dekonsentrasi, dan tugas pembantuan.

Dalam menyelenggarakan tugas pokok tersebut, Badan Perpustakaan dan Kearsipan Daerah Provinsi Jawa Barat mempunyai fungsi sebagai berikut :

1. Penyelenggaraan perumusan dan penetapan kebijakan teknis bidang perpustakaan dan kearsipan daerah.
2. Penyelenggaraan pemberian dukungan atas penyelenggaraan pemerintahan daerah bidang perpustakaan dan kearsipan daerah.

2.1.5 Struktur Organisasi

Struktur organisasi merupakan tata cara mengatur hubungan kerja setiap anggota, yaitu menentukan batas wewenang tugas beserta tanggung jawab masing-masing pegawai. Berdasarkan jabatan yang dipegang masing-masing dengan susunan organisasi akan terlihat dimana kedudukan seseorang dengan kegiatan dan fungsi yang telah ditentukan. Gambar 2.1 menjelaskan struktur organisasi di Badan Perpustakaan dan Kearsipan Daerah.



Gambar 2.1 Struktur Organisasi Badan Perpustakaan dan Kearsipan Daerah

2.1.6 Deskripsi Jabatan

Pembagian tugas dalam suatu instansi /lembaga/perusahaan/organisasi sangat penting, supaya terjadi keharmonisan dalam mencapai suatu tujuan.

Berikut ini terdapat deskripsi jabatan yang berlaku di Badan Perpustakaan dan Kearsipan Daerah Jawa Barat :

A. Tugas Kepala

1. Memimpin, mengatur, membina dan mengendalikan pelaksanaan tugas pokok dan fungsi badan.
2. Menetapkan kebijaksanaan teknis perancangan pengolahan perpustakaan.
3. Menetapkan program kerja dan rencana kegiatan perpustakaan.
4. Menyelenggarakan fasilitas dalam bidang perpustakaan meliputi pengembangan bahan pustaka, layanan dan pembinaan.
5. Memberikan saran, pertimbangan dan rekomendasi sebagai bahan pengambilan kebijakan gubernur.

B. Tugas Sekretaris

1. Mengkoordinasikan penyusunan program, evaluasi dan pelaporan tugas badan.
2. Melaksanakan pengaturan pengelolaan administrasi kepegawaian.
3. Mengkoordinasikan pengaturan pengelolaan administrasi keuangan rutin.
4. Melaksanakan pengelolaan urusan rumah tangga dan perlengkapan.
5. Mengkoordinasikan penyusunan perencanaan strategis dan laporan akuntabilitas kinerja kerja.
6. Melaksanakan pembinaan kelembagaan dan ketatalaksanaan pada unit kerja badan dilingkungan badan.
7. Melaksanakan perumusan bahan rancangan pendokumentasian perundang undangan, pengelolaan perpustakaan dan hubungan masyarakat.
8. Melaksanakan pengendalian administrasi keuangan belanja rutin.
9. Melaksanakan pengelolaan naskah dinas dan kearsipan.

10. Melaksanakan koordinasi dengan unit kerja terkait.
11. Melaksanakan evaluasi dan pelaporan.

C. Tugas Bidang Pengembangan Bahan Pustaka

1. Melaksanakan perumusan program kerja Bidang Pengembangan Bahan Pustaka.
2. Melaksanakan fasilitas kegiatan pengembangan bahan pustaka.
3. Melaksanakan perumusan bahan koordinasi dalam pelaksanaan kegiatan pengembangan literatur sekunder.
4. Melaksanakan perumusan bahan koordinasi dalam pelaksanaan kegiatan pengembangan bahan pustaka.
5. Melaksanakan koordinasi dengan unit kerja terkait.
6. Melaksanakan evaluasi pelaporan.

D. Tugas Bidang Layanan

1. Melaksanakan rencana dan program kerja.
2. Melaksanakan fasilitasi kegiatan layanan perpustakaan.
3. Melaksanakan perumusan bahan koordinasi dalam pelaksanaan kegiatan layanan perpustakaan.
4. Melaksanakan perumusan bahan koordinasi dalam pelaksanaan kegiatan otomasi perpustakaan.
5. Melaksanakan perumusan bahan koordinasi dalam dalam pelaksanaan preservasi bahan pustaka.
6. Melaksanakan koordinasi dengan unit kerja terkait.
7. Melaksanakan evaluasi pelaporan.

E. Tugas Bidang Pembinaan

1. Melaksanakan penyusunan program kerja bidang pembinaan.

2. Melaksanakan fasilitasi kegiatan pembinaan.
3. Melaksanakan penyusunan bahan koordinasi dalam pelaksanaan kegiatan pembinaan.
4. Melaksanakan penyusunan bahan koordinasi dalam pelaksanaan kegiatan pembinaan sumber daya manusia.
5. Melaksanakan penyusunan bahan koordinasi dalam pelaksanaan kegiatan pembinaan kelembagaan perpustakaan.
6. Melaksanakan penyusunan bahan koordinasi dalam pelaksanaan kegiatan kajian perpustakaan.
7. Melaksanakan koordinasi dengan unit kerja terkait.
8. Menyelenggarakan evaluasi dan pelaporan.

F. Tugas Subbidang layanan dan otomasi perpustakaan

1. Sub bidang layanan dan otomasi perpustakaan mempunyai tugas pokok melaksanakan penyusunan bahan kebijakan teknis dan fasilitasi otomasi perpustakaan.
2. Melaksanakan pengelolaan pangkalan data perpustakaan, pembinaan dan pengembangan otomasi perpustakaan, dan pengelolaan internet.

2.2 Information Retrieval (IR)

IR atau Pencarian Informasi dapat didefinisikan sebagai upaya untuk menemukan materi (biasanya dokumen) yang bersifat tidak terstruktur (biasanya teks) yang memenuhi kebutuhan informasi dari jumlah data yang sangat besar (biasanya disimpan dalam komputer).

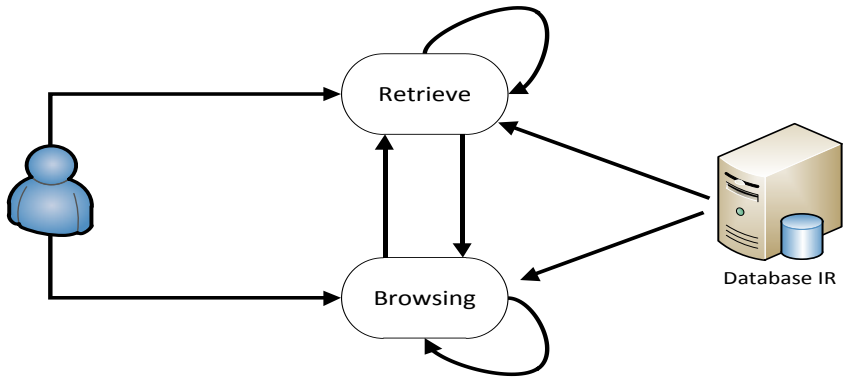
Salah satu pengertian dai IR menurut Bill Frakes dan Ricardo Yates[14] adalah "sub bidang dari ilmu komputer yang mempelajari tentang pengumpulan data dan temu kembali dokumen". Dalam perkembangan selanjutnya IR dikembangkan menjadi automated IR *systems* yang menangani temu kembali data secara otomatis dalam jumlah data yang besar.

Tujuan dari sistem IR adalah memenuhi kebutuhan informasi pengguna dengan me-retrieve semua dokumen yang mungkin relevan, pada waktu yang sama me-retrieve sesedikit mungkin dokumen yang tidak relevan.Sistem IR yang baik memungkinkan pengguna menentukan secara cepat dan akurat apakah isi dari dokumen yang diterima memenuhi kebutuhannya. Agar representasi dokumen lebih baik, dokumen-dokumen dengan topik atau isi yang mirip dikelompokkan bersama-sama [13].

Keefektifan dari temu kembali informasi yang diinginkan tergantung pada dua hal mendasar yaitu perilaku pengguna dan logical view dari sistem temu kembali.

2.2.1 Prilaku Pengguna

Dalam sistem IR, pengguna menterjemaahkan kebutuhan informasinya ke dalam bentuk kata *query*, dengan *query* ini maka sistem akan melakukan pencarian ke dalam kumpulan dokumen. Kualitas informasi yang dihasilkan dari sistem IR, secara langsung dipengaruhi dua hal yaitu penyajian dokumen secara logikal oleh sistem dan prilaku pengguna, prilaku pengguna maksudnya adalah berhubungan dengan kegiatan pengguna menentukan *query* yang sesuai dengan informasi yang diinginkan. Interaksi antara pengguna dengan sistem IR melalui aktifitas yang berbeda dapat digambarkan seperti dibawah ini.



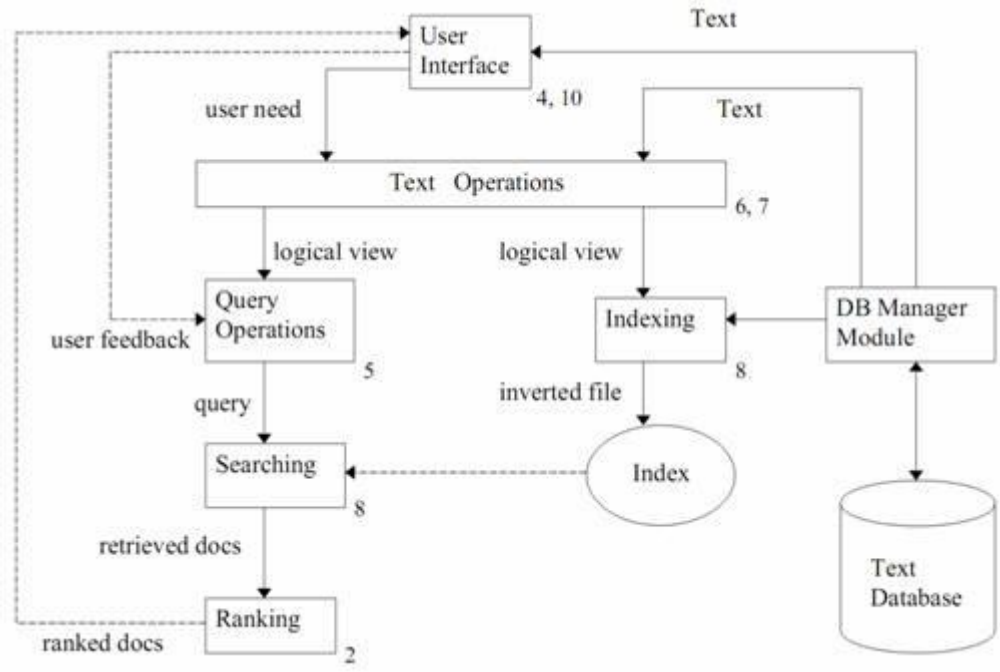
Gambar 2.2 Interkasi Sistem IR[28]

2.2.2 View Dokumen

Sistem basis data saat ini memungkinkan menyimpan representasi dokumen dalam bentuk koleksi keseluruhan kata-kata yang dikandungnya. Dalam hal ini dikatakan bahwa sistem IR mengadopsi *full text logical view*. Dengan banyaknya dokumen yang harus diproses, maka akan semakin besar kapaistas database yang diperlukan. *Full text* merupakan penampakan paling lengkap dari suatu dokumen tetapi penggunaannya membutuhkan biaya komputasi tinggi.

Untuk mengatasi hal tersebut, sistem harus dapat mereduksi kata yang disimpan dan mentransformasi logical view dari full text menjadi bentuk indeks. Hal ini dilakukan dengan meng-eliminasi stoplist (kata-kata yang terlalu umum seperti kata sandang, kata sambung, kata ganti, dll) dan melakukan proses stemming (pengubahan bentuk imbuhan ke bentuk dasar). Lebih jauh lagi adalah penggunaan metode-metode kompresi teks.

2.2.3 Arsitektur Sistem IR



Gambar 2.3 Arsitektur Sistem IR [25]

Secara garis besar arsitektur sistem IR diperlihatkan pada gambar II.2. Ada dua pekerjaan yang ditangani oleh sistem ini, yaitu melakukan pre-processing terhadap database dan kemudian menerapkan metode tertentu untuk menghitung kedekatan (relevansi atau similarity) antara dokumen di dalam database yang telah dipreprocess dengan query pengguna. Pada tahapan preprocessing, sistem yang berurusan dengan dokumen semi-structured biasanya memberikan tag tertentu pada term-term atau bagian dari dokumen; sedangkan pada dokumen tidak terstruktur proses ini dilewati dan membiarkan term tanpa imbuhan tag. Query yang dimasukkan pengguna dikonversi sesuai aturan tertentu untuk mengekstrak term-term penting yang sejalan dengan termterm yang sebelumnya telah diekstrak dari dokumen dan menghitung relevansi antara query dan dokumen berdasarkan pada term-term tersebut. Sebagai hasilnya, sistem mengembalikan suatu daftar dokumen terurut descending (ranking) sesuai nilai kemiripannya dengan query pengguna [15].

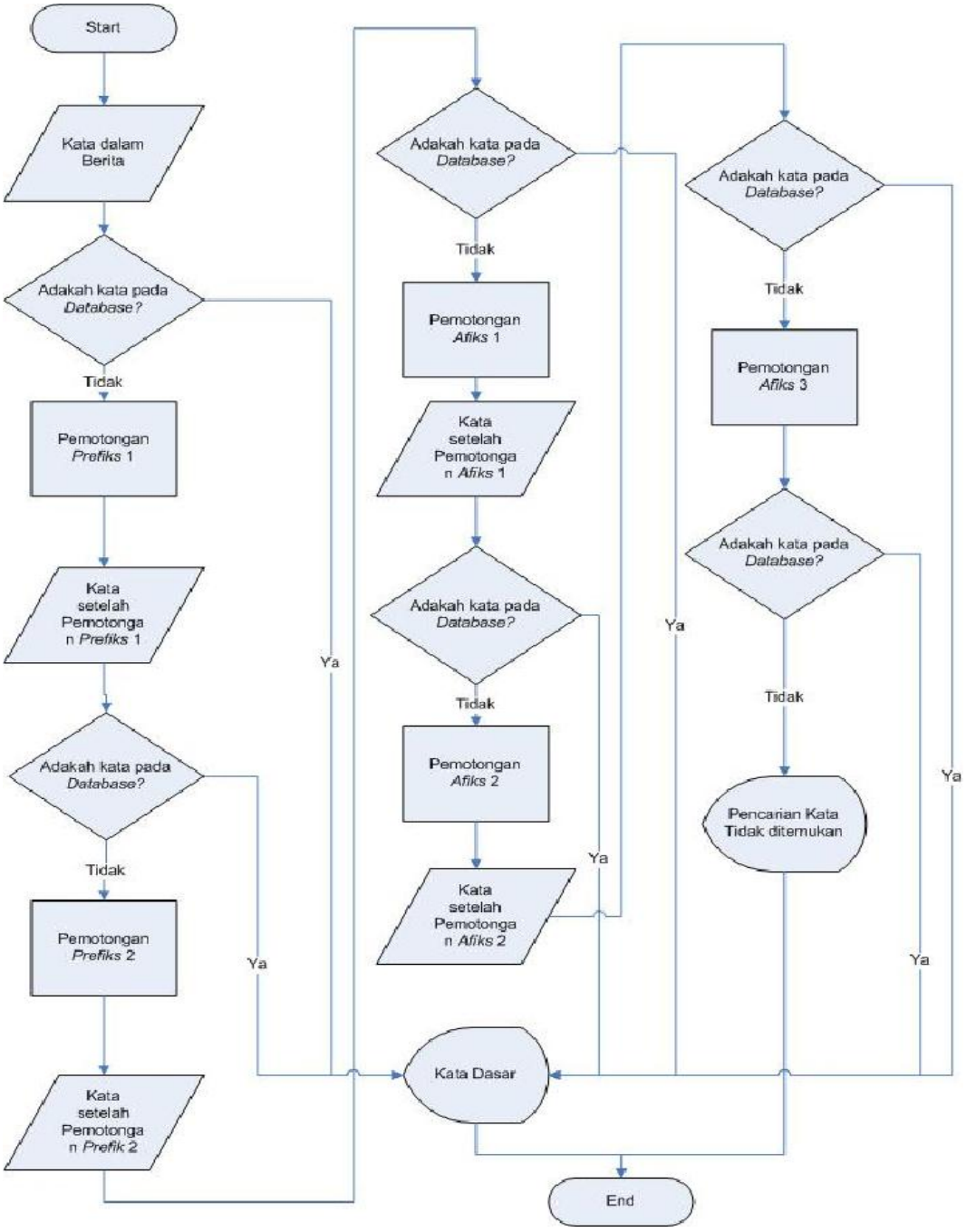
Setiap dokumen (termasuk *query*) direpresentasikan menggunakan *model bag-of-words* yang mengabaikan urutan dari kata-kata di dalam dokumen, struktur sintaktis dari dokumen dan kalimat. Dokumen ditransformasi ke dalam suatu "tas" berisi kata-kata independen. *Term* disimpan dalam suatu database pencarian khusus yang ditata sebagai sebuah *inverted index*. Index ini merupakan konversi dari dokumen asli yang mengandung sekumpulan kata ke dalam daftar kata yang berasosiasi dengan dokumen terkait dimana kata-kata tersebut muncul.

Algoritma Nazief & Adriani sebagai algoritma stemming untuk teks berbahasa Indonesia yang memiliki kemampuan prosentase keakuratan (presisi) lebih baik dari algoritma lainnya. Algoritma ini sangat dibutuhkan dan menentukan dalam proses IR dalam dokumen Indonesia[24].

2.2.4 *Stemming*

Stemming merupakan suatu proses yang terdapat dalam sistem IR yang mentransformasi kata-kata yang terdapat dalam suatu dokumen ke kata-kata akarnya (*root word*) dengan menggunakan aturan-aturan tertentu. Sebagai

contoh, kata bersama, kebersamaan, menyamai, akan distem ke *root word*-nya yaitu “sama”. Proses stemming pada teks ber Bahasa Indonesia berbeda dengan *stemming* pada teks berbahasa Inggris. Pada teks berbahasa Inggris, proses yang diperlukan hanya proses menghilangkan sufiks. Sedangkan pada teks berbahasa Indonesia, selain sufiks, prefiks, dan konfiks juga dihilangkan[24].



Gambar 2.4 *Flow Chart* Algoritma Nazief & Adriani [23]

2.2.5 Pembuatan *Index*

Pembuatan *index* dari koleksi dokumen merupakan tugas pokok pada tahapan *preprocessing* di dalam IR. Kualitas index mempengaruhi efektifitas dan efisiensi sistem IR [16]. *Index* dokumen adalah himpunan *term* yang menunjukkan isi atau topik yang dikandung oleh dokumen. *Index* akan membedakan suatu dokumen dari dokumen lain yang berada di dalam koleksi. Ukuran *index* yang kecil dapat memberikan hasil buruk dan mungkin beberapa item yang relevan terabaikan. *Index* yang besar memungkinkan ditemukan banyak dokumen yang relevan tetapi sekaligus dapat menaikkan jumlah dokumen yang tidak relevan dan menurunkan kecepatan pencarian [11].

Pembuatan inverted index harus melibatkan konsep linguistic processing yang bertujuan mengekstrak term-term penting dari dokumen yang direpresentasikan sebagai bag-ofwords. Ekstraksi term biasanya melibatkan dua operasi utama berikut [15]:

1. Penghapusan *stop-words*. *Stop-word* didefinisikan sebagai term yang tidak berhubungan (irrelevant) dengan subyek utama dari database meskipun kata tersebut sering kali hadir di dalam dokumen. Contoh beberapa daftar *stop-word*.

Tabel 2.1 Daftar *Stop-word*

Daftar *Stop Word*	**Daftar *Stop Word***
Yang	sudah
Mampu	tetapi
Tentang	bisa
Di	melakukannya
Setelah	lakukan
Semua	memang
Hampir	baik
Juga	lain
Am	pernah

Antara	setiap
Dan	untuk
Ada	dari
Seperti	mendapatkan
Jadi	punya
Karena	telah
Of	Mr
Mrs	…, dll

Stop-words termasuk pula beberapa kata tertentu yang didefinisikan terkait dengan topik database, misal pada database yang menampung daftar karya tulis (paper) penelitian terkait dengan heart diseases, maka kata heart dan disease sebaiknya dihapus.

2. Salah satu teknik untuk meningkatkan performa sistem IR adalah pencarian berdasarkan variasi morfologi kata. Salah satu contohnya untuk kata ’siap’ akan lebih mudah dikenali dalam suatu dokumen yang besar dibandingkan kata ‘mempersiapkan’. *Stemming* digunkan untuk mencari kata dasar dari bentuk berimbuhan, misalkan untuk kata lari, pelari, berlari, dilarikan, melarikan, semuanya akan dihitung menjadi satu *term* dengan frekuensi lima, bukan menjadi lima *term* berbeda dengan frekuensi masing-masing satu.

2.3 Vector Space Model (VSM)

VSM adalah model sistem temu balik informasi yang mengibaratkan masing-masing *query* dan dokumen sebagai sebuah vektor n-dimensi. Tiap dimensi pada vektor tersebut diwakili oleh satu *term*. *Term* yang digunakan biasanya berpatokan kepada *term* yang ada pada *query*, sehingga *term* yang ada pada dokumen tetapi tidak ada pada *query* biasanya diabaikan.

Secara umum prosedur VSM dibagi menjadi tiga tahap. Tahap pertama adalah *document indexing*, kumpulan *term* yang telah diekstrak dari teks

dokumen. Tahap kedua yaitu pembobotan dari *term* yang sudah di-indeks untuk meningkatkan kualitas pencarian. Tahap terkahir, rangking dokumen berdasarkan tingkat kemiripan antara *query* dengan dokumen[8].

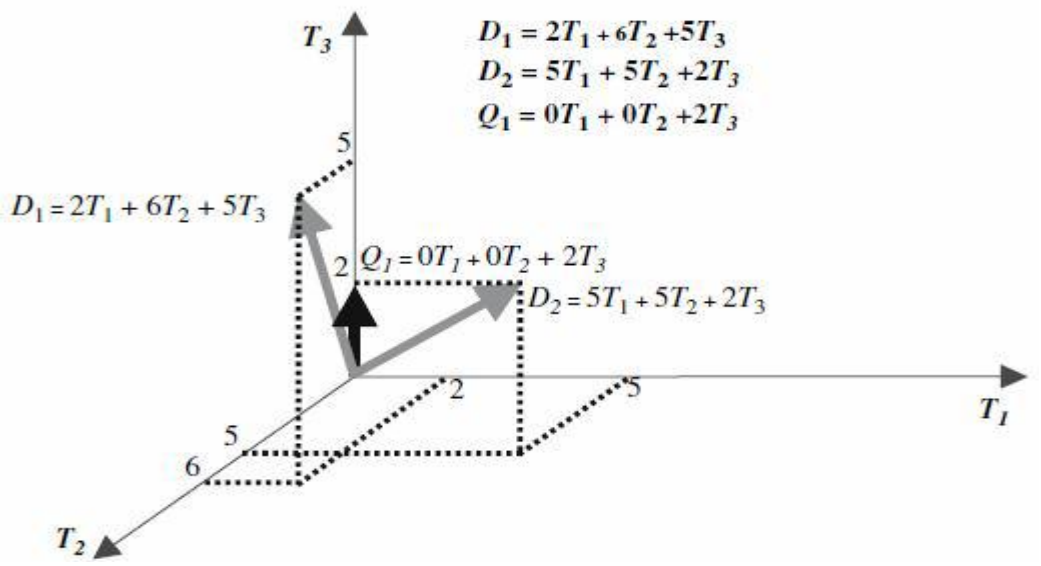
Pada *Vector Space Model* ini :

1 *Vocabulary* merupakan kumpulan semua *term* berbeda yang tersisa dari dokumen setelah preprocessing dan mengandung *t term index*. *Term-term* ini membentuk suatu ruang vektor.

2 Setiap *term i* di dalam dokumen atau query *j*, diberikan suatu bobot (weight) bernilai *real wij*.

3 Dokumen dan *query* diekspresikan sebagai vektor *t* dimensi *dj* = (*w1*, *w2*, ..., *wtj*) dan terdapat n dokumen di dalam koleksi, yaitu *j* = 1, 2, ..., n.

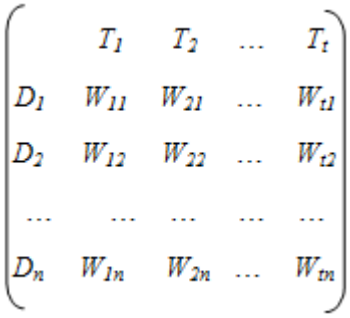
Contoh dari model ruang vektor tiga dimensi untuk dua dokumen D1 dan D2, satu query pengguna Q1, dan tiga term T1, T2 dan T3 diperlihatkan pada gambar 2.5.



Gambar 2.5 *Vector Space Model*[15]

Dalam model ruang vektor, koleksi dokumen direpresentasikan oleh matriks *term* document (atau matriks *term-frequency*). Setiap sel dalam matriks bersesuaian dengan bobot yang diberikan dari suatu *term* dalam dokumen yang

ditentukan. Nilai nol berarti bahwa *term* tersebut tidak hadir di dalam dokumen[15].



Gambar 2.6 Matriks *Term Document*

2.3.1 *Indexing*

Pengindeksan / *indexing* dilakukan untuk mendapatkan bobot dari setiap term dalam dokumen. Penghitungan bobot tersebut dilakukan dengan melakukan perhitungan terhadap *Term Frequency* (Tf) dan *Document Frequency* (Df) dari tiap term yang terdapat di koleksi dokumen, nilai Df selanjutnya akan diproses menjadi nilai *Inverse Document Frequency* (Idf) yang akan digunakan dalam perhitungan bobot term[12].

Keberhasilan dari model VSM ini ditentukan oleh skema pembobotan terhadap suatu term baik untuk cakupan lokal maupun global, dan faktor normalisasi [20]. Pembobotan lokal hanya berpedoman pada frekuensi munculnya term dalam suatu dokumen dan tidak melihat frekuensi kemunculan term tersebut di dalam dokumen lainnya. Pendekatan dalam pembobotan lokal yang paling banyak diterapkan adalah term frequency (tf) meskipun terdapat skema lain seperti pembobotan biner, augmented normalized tf, logaritmik tf dan logaritmik alternatif.

Idf dapat ditentukan dengan menggunakan rumus:

$$idf = log\left[\frac{d}{df}\right] \quad (1)$$

Dengan ketentuan :

Idf = *inverse document frequency*

d = *total dokumen*

df = jumlah dokumen yang mengandung term

Bobot dari setiap *term* dapat dihitung dengan rumus *w = tf * idf*

2.3.2 Normalisasi

Faktor normalisasi digunakan untuk menormalkan vektor dokumen sehingga proses retrieval tidak terpengaruh oleh panjang dari dokumen. Normalisasi ini diperlukan karena dokumen panjang biasanya mengandung perulangan *term* yang sama sehingga menaikkan frekuensi *term* (tf). Dokumen panjang juga mengandung banyak *term* yang berbeda sehingga menaikkan ukuran kemiripan antara query dengan dokumen tersebut, meningkatkan peluang di-retrievenya dokumen yang lebih panjang.

Normalisasi yang digunakan adalah normalisasi kosinus. Berdasarkan [22] rumus normalisasi kosinus yaitu :

$$w(word_i) = \frac{w\ (word_i)}{\sqrt{w^2(word_1) + w^2(word_2) + \cdots + w^2(word_n)}} \quad (2)$$

Dengan *W* adalah bobot dari *query* dan dokumen.

2.4 Ukuran Kemiripan

Model ruang vektor dan pembobotan *tf-idf* digunakan untuk merepresentasikan nilai numerik dokumen sehingga kemudian dapat dihitung kedekatan antar dokumen. Semakin dekat dua vektor di dalam suatu VSM maka semakin mirip dua dokumen yang diwakili oleh vektor tersebut. Kemiripan antar dokumen dihitung menggunakan suatu fungsi ukuran kemiripan (*similarity measure*). Ukuran ini memungkinkan perankingan dokumen sesuai dengan kemiripan relevansinya terhadap *query*. Setelah dokumen diranking, sejumlah tetap dokumen *top-scoring* dikembalikan kepada pengguna.

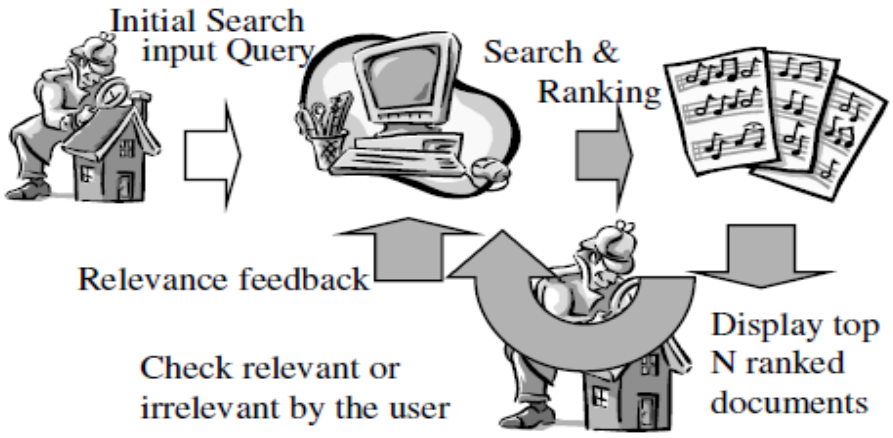
Cosine Similarity tidak hanya digunakan untuk menghitung normalisasi panjang dokumen tapi juga menjadi salah satu ukuran kemiripan yang popular[21]. Ukuran ini menghitung nilai kosinus sudut antara dua vektor. Jika terdapat dua vektor dokumen *d* dan *query q*, serta *t term* diekstrak dari koleksi dokumen maka nilai kosinus antara *d* dan *q* didefinisikan sebagai [15] :

$$similarity\ (\vec{d}_j, \vec{q}) = \frac{\vec{d}_j . \vec{q}}{|\vec{d}_j| . |\vec{q}|} = \frac{\sum_{i=1}^{t} (w_{ij} \cdot w_{iq})}{\sqrt{\sum_{i=1}^{t} w_{ij}^2 \cdot \sum_{i=1}^{t} w_{iq}^2}} \quad (3)$$

2.5 Peningkatan Kualitas Pencarian

Rancangan dasar dari sistem IR dapat ditingkatkan untuk menaikkan presisi dan *recall* serta memperbaiki matriks *term-document*. Isu pertama sering diselesaikan menggunakan mekanisme *relevance feedback*. Beberapa *term* ditambahkan ke dalam *query* awal agar dapat menemukan dokumen yang lebih relevan. *Relevance feedback* dapat dikerjakan secara manual maupun otomatis. Pada pendekatan manual, pengguna mengidentifikasi dokumen yang relevan dan *term* baru dipilih secara manual atau otomatis. Pada pendekatan otomatis, dokumen relevan diidentifikasi menggunakan dokumen *top-ranked*, kemudian *term-term* baru dipilih secara otomatis.

Takashi Onoda, dkk dalam penelitianya yang berjudul *Relevance Feedback Document Retrieval using Support Vector Machines*, telah berhasil melakukan peningkatan kualitas pencarian dengan memodifikasi vektor *query*. Gambar dibawah ini menjelaskan konsep *Relevance Feedback* untuk meningkatkan kualitas pencarian.



Gambar 2.7 Skema Umpan Balik dalam Pencarian Dokumen[9]

2.6 Support Vector Machines (SVM)

Support Vector Machines (SVM) adalah suatu teknik yang relatif baru (1992) untuk melakukan prediksi, baik dalam kasus klasifikasi maupun regresi, yang sangat populer belakangan ini. SVM berada dalam satu kelas dengan *Artificial Neural Network* (ANN) dalam hal fungsi dan kondisi permasalahan yang bisa diselesaikan. Keduanya masuk dalam kelas *supervised learning*. Baik para ilmuwan maupun praktisi telah banyak menerapkan teknik ini dalam menyelesaikan masalah-masalah nyata dalam kehidupan sehari-hari. Baik dalam masalah *gene expression analysis*, finansial, cuaca hingga di bidang kedokteran. Terbukti dalam banyak implementasi, SVM memberi hasil yang lebih baik dari ANN, terutama dalam hal solusi yang dicapai. ANN menemukan solusi berupa *local optimal* sedangkan SVM menemukan solusi yang *global optimal*. Tidak heran bila solusi ANN dari setiap training selalu berbeda. Hal ini disebabkan solusi *local optimal* yang dicapai tidak selalu sama. SVM selalu mencapi solusi yang sama untuk setiap running. Dalam teknik ini, SVM berusaha untuk menemukan *hyperplane* (pemisah/*classifier*) yang optimal yang bisa memisahkan dua set data dari dua kelas yang berbeda.

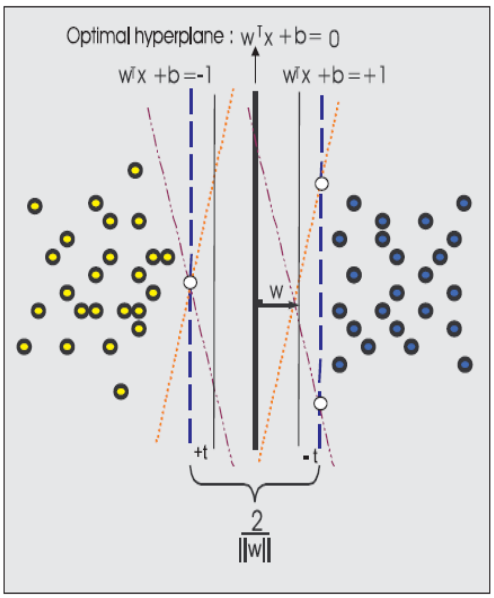
Dalam hal ini fungsi pemisah yang dicari adalah fungsi linier. Fungsi ini bisa didefinisikan sebagai :

$g(x) := sgn(f(x))$ (4)
dengan $f(x) = w^T x + b$

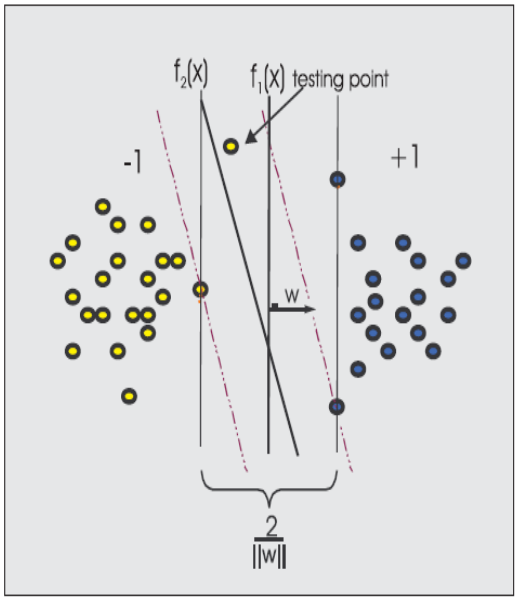
Dimana x, $w \in R^n$ dan $b \in R$. Masalah klasifikasi ini bisa dirumuskan sebagai berikut: kita ingin menemukan set parameter (w, b) sehingga $f(x_i) =< w,x > + b= y_i$ untuk semua i.

Dalam teknik SVM berusaha menemukan fungsi *hyperplane* terbaik diantara fungsi yang tidak terbatas jumlahnya untuk memisahkan dua macam obyek. *Hyperplane* terbaik adalah *hyperplane* yang terletak di tengah-tengah antara dua set obyek dari dua kelas. Mencari *hyperplane* terbaik ekuivalen dengan memaksimalkan margin atau jarak antara dua set obyek dari kelas yang berbeda. Jika $wx_1 + b = +1$ adalah *hyperplane*-pendukung (*supporting hyperplane*) dari kelas +1 ($wx_1 + b = +1$) dan $wx_2 + b = -1$ hyperplane-pendukung dari kelas -1 ($wx_2 + b = -1$) margin antara dua kelas dapat dihitung dengan mencari jarak antara kedua hyperplane-pendukung dari kedua kelas. Secara spesifik, margin dihitung dengan cara berikut ($wx_1 + b = +1$) - ($wx_2 + b = -1$) => $w(x_1 – x_2) = 2$ => $(\frac{w}{||w||}(x_1 – x_2)) = \frac{2}{||w||}$. Gambar 2.8 memperlihatkan bagaimana SVM bekerja untuk menemukan suatu fungsi pemisah dengan margin yang maksimal. Untuk membuktikan bahwa memaksimalkan margin antara dua set obyek akan meningkatkan probabilitas pengelompokkan secara benar dari data testing. Pada dasarnya jumlah fungsi pemisah ini tidak terbatas banyaknya. Misalkan dari jumlah yang tidak terbatas ini kita ambil dua saja, yaitu $f_1(x_1)$ dan $f_2(x_2)$. Fungsi f_1 mempunyai margin yang lebih besar dari pada fungsi f_2. Setelah menemukan dua fungsi ini, sekarang suatu data baru masuk dengan keluaran −1. Kita harus mengelompokkan apakah data ini ada dalam kelas −1 atau +1 menggunakan fungsi pemisah yang sudah kita temukan. Dengan menggunakan f_1, kita akan kelompokkan data baru ini di kelas −1 yang berarti kita benar mengelompokkannya. Sekarang coba kita gunakan f_2, kita akan menempatkannya di kelas +1 yang berarti salah. Dari contoh sederhana ini kita lihat bahwa

memperbesar margin bisa meningkatkan probabilitas pengelompokkan suatu data secara benar.



Gambar 2.8 Mencari Fungsi Pemisah Optimal yang dapat Memisahkan Data Secara Linear[29]



Gambar 2.9 Menentukan Data Baru[29]

2.6.1 Formulasi Matematis

Secara matematika, formulasi problem optimisasi SVM untuk kasus klasifikasi linier di dalam *primal space* adalah

$$\min \quad \frac{1}{2}\,||w||^2 \tag{5}$$

Subject to

$y_i\,(wx_i + b) \geq 1,\ i = 1,\ldots,\ n$

dimana x_i adalah data input, y_i adalah keluaran dari data x_i, w, b adalah parameter-parameter yang kita cari nilainya. Formulasi di atas bertujuan meminimalkan fungsi tujuan (obyektif function) dari persamaan(5) atau memaksimalkan kuantitas $||w||^2$ atau w^Tw dengan memperhatikan pembatas y_i $(wx_i + b) \geq 1$. Bila output data $y_i = +1$, maka pembatas menjadi $(wx_i + b) \geq 1$. Sebaliknya bila $y_i = -1$, pembatas menjadi $(wx_i + b) \leq -1$.

Di dalam kasus yang tidak feasible (*infeasible*) dimana beberapa data mungkin tidak bisa dikelompokkan secara benar, formulasi matematikanya menjadi berikut :

$$min\,\frac{1}{2}\,||w||^2 + C\sum_{i=1}^{l} t_i \tag{6}$$

Subject to

$y_i\,(wx_i + b)\ + t_i \geq 1,\ t_i \geq 0,\ i = 1,\ldots,\ l$

dimana t_i adalah variabel *slack.* Dengan formulasi ini kita ingin memaksimalkan margin antara dua kelas dengan meminimalkan $||w||^2$. Formulasi ini berusaha meminimalkan kesalahan klasifikasi (*misclassification error*) yang dinyatakan dengan adanya variabel *slack* t_i, secara bersamaan formulasi tersebut memaksimalkan margin $\frac{1}{||w||}$. penggunaan variabel *slack* t_i adalah untuk mengatasi kasus ketidaklayakan (*infeasibility*) dari pembatas (*constraints*) $y_i\,(wx_i + b) \geq 1$ dengan cara memberi pinalti untuk data yang tidak memenuhi pembatas

tersebut. Untuk meminimalkan nilai t_i ini, kita berikan pinalti dengan menerapkan konstanta ongkos C. Vektor *w* tegak lurus terhadap fungsi pemisah: $wx + b = 0$. Konstanta *b* menentukan lokasi fungsi pemisah relatif terhadap titik asal.

Problem (3) adalah programa nonlinear. Ini bisa dilihat dari fungsi tujuan (*objective function*) yang berbentuk kuadrat. Untuk menyelesaikannya, secara komputasi agak sulit dan perlu waktu lebih panjang. Untuk membuat masalah ini lebih mudah dan efisien untuk diselesaikan, masalah ini bisa kita transformasikan ke dalam *dual space*. Untuk itu, pertama kita ubah problem (6) menjadi fungsi *Lagrangian* :

$$J(w, b, \alpha) = \frac{1}{2} w^T w - \sum_{i=1}^{l} \alpha[y_i(w^T x_i) - 1] \tag{7}$$

dimana variabel non-negatif α_i, dinamakan *lagrange multiplier*. Solusi dari problem optimisasi dengan pembatas seperti di atas ditentukan dengan mencari saddle point dari fungsi Lagrangian $J(w, b, \alpha)$. Fungsi ini harus diminimalkan terhadap variabel *w* dan b dan harus dimaksimalkan terhadap variabel α. Kemudian kita cari turunan pertama dari fungsi $J(w, b, \alpha)$ terhadap variabel *w* dan *b* dan kita samakan dengan 0. Dengan melakukan proses ini, kita akan mendapatkan dua kondisi optimalitas berikut:

Kondisi 1 :

$$\frac{\delta j(w, b, \alpha)}{\delta w} = 0$$

Kondisi 2 :

$$\frac{\delta j(w, b, \alpha)}{\delta b} = 0$$

Penerapan kondisi optimalitas 1 pada fungsi Lagrangian (7) akan menghasilkan

$$w = \sum_{i=1}^{l} \alpha_i y_i x_i \tag{8}$$

Penerapan kondisi optimalitas 2 pada fungsi *Lagrangian* (7) akan menghasilkan

$$\sum_{i=1}^{l} \alpha_i y_i = 0 \tag{9}$$

Menurut duality theorem [26]:

1 Jika problem primal mempunyai solusi optimal, maka problem dual juga akan mempunyai solusi optimal yang nilainya sama

2 Bila wo adalah solusi optimal untuk problem primal dan αo untuk problem dual, maka perlu dan cukup bahwa wo solusi layak untuk problem primal dan

$$\varphi(w_0) = J(w_0, b_0, \alpha_0) = min_w\, J(w, b, \alpha)$$

Setelah itu, jabarkan persamaan (4) sebagai berikut:

$$J(w, b, \alpha) = \frac{1}{2} w^T w - \sum_{i=1}^{l} \alpha_i y_i w^T x_i - b \sum_{i=1}^{l} \alpha_i y_i + \sum_{i=1}^{l} \alpha_i \tag{10}$$

Menurut kondisi optimalitas ke dua dalam (9), term ketiga sisi sebelah kanan dalam persamaan di atas sama dengan 0. Dengan memakai nilainilai *w* di (8), maka didapat

$$w^T w = \sum_{i=1}^{l} \alpha_i y_i w^T x_i = \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j x_i^T x_j \tag{11}$$

maka persamaan 7 menjadi :

$$Q(\alpha) = \frac{1}{2} \sum_{1,j=1}^{l} y_i y_j \alpha_i \alpha_j x_i^T x_j \tag{12}$$

Selanjutnya kita dapatkan formulasi dual dari problem (6):

$$\max \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j}^{l} y_i y_j \alpha_i \alpha_j x_i^T x_j \quad (13)$$

Subject to $\sum_{i=1}^{l} \alpha_i y_i = 0$

$0 \leq \alpha_i$, $i = 1, ..l$,

Dengan dot product $x_i x_j$ sering diganti dengan simbol *K*. *K* adalah matrik kernel. Formulasi (10) adalah *quadratic programming* (QP) dengan pembatas (*constraint*) linier. Melatih SVM ekuivalen dengan menyelesaikan problem *convex optimization*. Karena itu solusi dari SVM adalah unik (dengan asumsi bahwa k adalah *positive definite*) dan global optimal. Ambil,

$$f(x) = \sum_{i=1}^{l} y_i \alpha_i^* k(x_i, x) + b^* \quad (14)$$

Fungsi pemisah optimal adalah

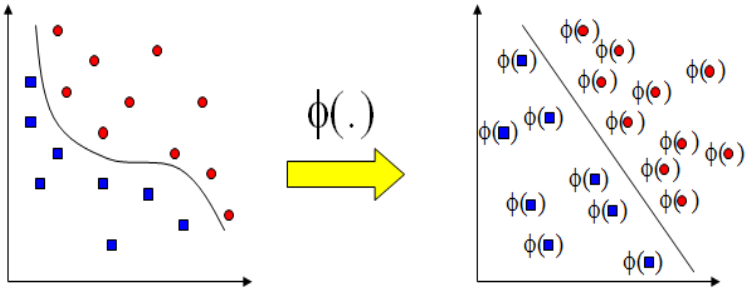
$$g(x) = sign(\sum_{i=1}^{l} y_i \alpha_i^* k(x_i, x)) + b^* \quad (15)$$

dimana α_i^*, i =1,..*l* adalah solusi optimal dari problem (13) dan b^* dipilih sehingga $y_i f(x_i) = 1$ untuk sembarang i dengan $C > \alpha_i^* > 0$. Data x_i dimana $\alpha_i^* > 0$ dinamakan support vector dan menyatakan data training yang diperlukan untuk mewakili fungsi keputusan yang optimal. Dalam gambar 1, sebagai contoh, 3 titik berwarna putih menyatakan support vector. Untuk mengatasi masalah ketidaklinieran (*nonlinearity*) yang sering terjadi dalam kasus nyata, kita bisa menerapkan metode kernel. Metode kernel memberikan pendekatan alternatif dengan cara melakukan *mapping* data x dari *input space* ke *feature space F* melalui suatu fungsi φ sehingga $\varphi : x \rightarrow \varphi(x)$. Karena itu suatu titik *x* dalam *input space* menjadi $\varphi(x)$ dalam *feature space*.

2.6.2 Metode Kernel

Banyak teknik data mining atau *machine learning* yang dikembangkan dengan asumsi kelinieran. Sehingga algorithma yang dihasilkan terbatas untuk kasus-kasus yang linier. Karena itu, bila suatu kasus klasifikasi memperlihatkan ketidaklinieran, algorithma seperti perceptron tidak bisa mengatasinya. Secara umum, kasus-kasus di dunia nyata adalah kasus yang tidak linier. Sebagai contoh, perhatikan Gambar 3. Data ini sulit dipisahkan secara linier. Metode kernel [5] adalah salah satu untuk mengatasinya. Dengan metoda kernel suatu data *x* di input space dimapping ke feature space F dengan dimensi yang lebih tinggi melalui map φ sebagai berikut $\varphi : x \rightarrow \varphi(x)$. Karena itu data *x* di input space menjadi φ (*x*) di *feature space*.

Sering kali fungsi $\varphi(x)$ tidak tersedia atau tidak bisa dihitung. tetapi *dot product* dari dua vektor dapat dihitung baik di dalam input space maupun di feature space. Dengan kata lain, sementara $\varphi(x)$ mungkin tidak diketahui, dot product $< \varphi(x_1), \varphi(x_2) >$ masih bisa dihitung di feature space. Untuk bisa memakai metoda kernel, pembatas (constraint) perlu diekspresikan dalam bentuk dot product dari vektor data x_i. Sebagai konsekuensi, pembatas yang menjelaskan permasalahan dalam klasifikasi harus diformulasikan kembali sehingga menjadi bentuk dot product. Dalam feature space ini dot product $< . >$ menjadi $<\varphi(x_1), \varphi(x_2)' >$. Suatu fungsi kernel, $k(x, x')$, bisa untuk menggantikan dot product $<\varphi(x_1), \varphi(x_2)' >$ Kemudian di feature space, kita bisa membuat suatu fungsi pemisah yang linier yang mewakili fungsi nonlinear di input space. Gambar 4 mendeskripsikan suatu contoh feature mapping dari ruang dua dimensi ke feature space dua dimensi. Dalam *input space*, data tidak bisa dipisahkan secara linier, tetapi kita bisa memisahkan di *feature space*. Karena itu dengan memetakan data ke *feature space* menjadikan tugas klasifikasi menjadi lebih mudah [5].



Gambar 2.10 Mapping Data dari *Input Space* ke *Feature Space*[29]

Fungsi kernel yang biasanya dipakai dalam literatur SVM [7]:

1. Linear : $x^T x$,
2. *Polynomial* : $(x^T x + 1)^p$,
3. *Radial basis function* : $\exp(-\frac{||x - x^i||^2}{2\sigma^2})$
4. *Tangent hyperbolic* (*sigmoid*) : $tanh(\beta x^T x_i + \beta_1)$, dimana $\beta, \beta_1 \in R$

Fungsi kernel yang legitimate diberikan oleh *Teori Mercer* [6] dimana fungsi itu harus memenuhi syarat: kontinus dan positive definite. Lebih mudah menemukan fungsi kernel daripada mencari map φ seperti apa yang tepat untuk melakukan *mapping* dari *input space* ke *feature space*. Pada penerapan metode kernel, tidak perlu tahu map apa yang digunakan untuk satu per satu data, tetapi lebih penting mengetahui bahwa dot produk dua titik di *feaure space* bisa digantikan oleh fungsi kernel.

2.7 MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: database management system) atau DBMS yang multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.

Tidak sama dengan proyek-proyek seperti Apache, dimana perangkat lunak dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia MySQL AB, dimana memegang hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah: David Axmark, Allan Larsson, dan Michael "Monty" Widenius.

MySQL memiliki beberapa keistimewaan, antara lain :

1. **Portabilitas.** MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.
2. **Perangkat lunak sumber terbuka.** MySQL didistribusikan sebagai perangkat lunak sumber terbuka, dibawah lisensi GPL sehingga dapat digunakan secara gratis.
3. **Multi-user.** MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
4. '**Performance tuning**', MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.
5. **Ragam tipe data**. MySQL memiliki ragam tipe data yang sangat kaya, seperti signed / unsigned integer, float, double, char, text, date, timestamp, dan lain-lain.
6. **Perintah dan fungsi**. MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah Select dan Where dalam perintah (query).
7. **Keamanan**. MySQL memiliki beberapa lapisan keamanan seperti level subnetmask, nama host, dan izin akses user dengan sistem perizinan yang mendetail serta sandi terenkripsi.

8. **Skalabilitas dan pembatasan**. MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (records) lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.
9. **Konektivitas**. MySQL dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, Unix soket (UNIX), atau Named Pipes (NT).
10. **Lokalisasi**. MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meski pun demikian, bahasa Indonesia belum termasuk di dalamnya.
11. **Antar muka**. MySQL memiliki antar muka (interface) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (Application Programming Interface).
12. **Klien dan peralatan**. MySQL dilengkapi dengan berbagai peralatan (tool)yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk online.
13. **Struktur tabel**. MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan basis data lainnya semacam PostgreSQL ataupun Oracle.

2.8 PHP

PHP berawal dari skrip *Perl/CGI* yang dibuat oleh seorang pengembang perangkat lunak bernama Rasmus Lerdorf untuk menghitung jumlah pengunjung *homepage*-nya. Karena banyaknya pengunjung yang meminta skrip tersebut, Lerdorf akhirnya membagi-bagikan skrip buatannya yang diberi nama *Personal Home Page (PHP)*.

Banyaknya permintaan membuat Lerdorf terus mengembangkan skripnya. Beberapa orang akhirnya bergabung membentuk tim untuk mengembangkan PHP. Sejak itu PHP berkembang pesat dengan banyak fungsi baru yang ditambahkan. Kepanjangan dari PHP kini berubah menjadi PHP: Hypertext Preprocessor.

Ada tiga macam penggunaan PHP:

1. *Server-side scripting*. Ini merupakan jenis penggunaan yang paling banyak dilakukan pengguna PHP. Untuk menggunakannya, dibutuhkan tiga hal: PHP *parser*, aplikasi *web server* yang terkoneksi dengan instalasi PHP, dan aplikasi *web browser*.
2. *Command line scripting*. Pada penggunaan PHP jenis ini hanya dibutuhkan PHP parser.
3. Pembuatan aplikasi berbasis *desktop*. Pada penggunaan PHP jenis ini, dibutuhkan ekstensi tambahan *PHP-GTK*.

PHP memiliki empat kelebihan utama yang menarik minat banyak pengguna. Kelebihan utama PHP tersebut diringkas dalam 4P berikut:

1. *Practicality*. PHP dibuat dengan menitikberatkan pada kepraktisan. Hasilnya, PHP adalah bahasa pemrograman minimalis, dilihat dari segi kebutuhan pengguna dan kebutuhan sintaks.
2. *Power*. PHP memiliki banyak kemampuan, mulai dari kemampuan untuk terhubung dengan basis data, membuat halaman web dinamis, membuat dan memanipulasi berkas gambar, Flash dan PDF, berkomunikasi dengan bermacam protokol seperti IMAP dan POP3, dan masih banyak lagi.
3. *Possibility*. PHP dapat menyediakan lebih dari satu solusi untuk suatu masalah
4. *Price*. PHP selalu dirilis kepada publik tanpa ada batasan untuk penggunaan, modifikasi, atau redistribusi.

2.9 Cascading Style Sheets (CSS)

CSS adalah sebuah bahasa *style sheet* (lembar gaya) yang digunakan untuk mengatur tampilan dokumen yang ditulis dalam bahasa *markup*. *CSS Level 1* (CSS1) diperkenalkan pada tahun 1995 oleh *World Wide Web Consortium* (W3C) dan dimaksudkan untuk mengatasi keterbatasan HTML dalam hal keleluasaan pengaturan desain dan tampilan pada sebuah dokumen HTML. Setahun kemudian

CSS1 diberi status rekomendasi penuh oleh W3C yang juga mengatur spesifikasi CSS. Saat ini ada tiga level CSS, yaitu CSS *Level* 1 (*Recommendation*), CSS *Level* 2 (*Recommendation*), dan CSS *Level* 2 *Revision* 1 (*Candidate Recommendation*). Penggunaan CSS paling banyak untuk memformat halaman web yang ditulis dengan HTML dan XHTML. Walau demikian, CSS dapat dipergunakan untuk bahasa markup lain seperti SVG dan XUL.

2.10 XAMPP

XAMPP merupakan singkatan dari X (empat *operating system* apapun) Apache, MySQL, PHP, Perl. XAMPP merupakan tool yang menyediakan paket perangkat lunak ke dalam satu buah paket. Dalam paketnya sudah terdapat Apache (web server), MySQL (database), PHP (server side scripting), Perl, FTP server, phpMyAdmin dan berbagai pustaka bantu lainnya. Dengan menginstall XAMPP maka tidak perlu lagi melakukan instalasi dan konfigurasi web server Apache, PHP dan MySQL secara manual. XAMPP akan menginstalasi dan mengkonfigurasikannya secara otomatis.

2.11 Adobe Dreamweaver CS5

Dreamweaver merupakan sebuah aplikasi untuk merancang pembuatan *website*. Dreamweaver dibuat oleh perusahaan Macromedia sehingga dinamakan Macromedia Dreamweaver. Sejak Macromedia diakuisisi Adobe Inc., namanya berubah menjadi Adobe Dreamweaver. Versi pertama Dreamweaver dibawah Adobe adalah CS5, mengikuti versi rilisnya yang dipaketkan dalam Adobe Creative Suite 5.

Dreamweaver memiliki kelebihan dalam hal kemudahan penggunaan. Untuk pengguna awam, Dreamweaver menyediakan fungsi tampilan Design, sehingga pengguna dapat merancang tampilan halaman web dengan konsep WYSIWYG (*What You See Is What You Get*). Untuk pengguna tingkat lanjut, Dreamweaver menyediakan tampilan *Code* sehingga pengguna dapat merancang tampilan yang lebih lengkap menggunakan kode. Pengguna juga dimudahkan

dengan berbagai fasilitas yang dimiliki Dreamweaver seperti tag auto-completionuntuk penulisan kode HTML. Format yang didukung Dreamweaver juga cukup lengkap, mulai dari HTML, JavaScript, CSS, sampai XML.

2.12 *Unified Modelling Language* (UML)

UML adalah bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak.

UML adalah himpunan struktur danteknik untuk pemodelan desain program berorientasi objek (OOP) serta aplikasinya. UML adalah metodologi untuk mengembangkan sistem OOP dan sekelompok perangkat tool untuk mendukung pengembangan sistem tersebut UML mulai diperkenalkan oleh Object Management Group, sebuah organisasi yang telah mengembangkan model, teknologi, dan standar OOP sejak tahun 1980-an. Sekarang UML sudah mulai banyak digunakan oleh para praktisi OOP. UML merupakan dasar bagi perangkat (tool) desain berorientasi objek dari IBM.

UML adalah suatu bahasa yang digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan suatu sistem informasi. UML dikembangkan sebagai suatu alat untuk analisis dan desain berorientasi objek oleh Grady Booch, Jim Rumbaugh, dan Ivar Jacobson.Namun demikian UML dapat digunakan untuk memahami dan mendokumentasikan setiap sistem informasi.Penggunaan UML dalam industri terus meningkat. Ini merupakan standar terbuka yang menjadikannya sebagai bahasa pemodelan yang umum dalam industri peranti lunak dan pengembangan sistem.

UML menyediakan 10 macam diagram untuk memodelkan aplikasi berorientasi objek, yaitu:

1. *Use Case Diagram* untuk memodelkan proses bisnis.
2. *Conceptual Diagram* untuk memodelkan konsep-konsep yang ada di dalam aplikasi.

3. *Sequence Diagram* untuk memodelkan pengiriman pesan (message) antar objek.
4. *Collaboration Diagram* untuk memodelkan interaksi antar objek.
5. *State Diagram* untuk memodelkan perilaku objek di dalam sistem.
6. *Activity Diagram* untuk memodelkan perilaku userdan objek di dalam sistem.
7. *Class Diagram* untuk memodelkan struktur kelas.
8. *Objek Diagram* untuk memodelkan struktur objek.
9. *Component Diagram* untuk memodelkan komponen objek.
10. *Deployment Diagram* untuk memodelkan distribusi aplikasi.

Berikut akan dijelaskan 4 macam diagram yang paling sering digunakan dalam pembangunan aplikasi berorientasi objek, yaitu use case diagram, sequence diagram, collaboration diagram, dan class diagram.

2.12.1 *Use Case Diagram*

Use case diagram digunakan untuk memodelkan bisnis proses berdasarkan perspektif pengguna sistem. *Use case* diagram terdiri atas diagram untuk *use case* dan *actor*. *Actor* merepresentasikan orang yang akan mengoperasikan atau orang yang berinteraksi dengan sistem aplikasi.

Use case merepresentasikan operasi-operasi yang dilakukan oleh *actor*. *Use case* digambarkan berbentuk elips dengan nama operasi dituliskan didalamnya. Actor yang melakukan operasi dihubungkan dengan garis lurus ke *use case*.

2.12.2 *Sequence Diagram*

Sequence diagram menjelaskan secara detil urutan proses yang dilakukan dalam sistem untuk mencapai tujuan dari *use case.* Interaksi yang terjadi antar *class,* operasi apa saja yang terlibat, urutan antar operasi, dan informasi yang diperlukan oleh masing-masing operasi.

2.12.3 *Activity Diagram*

Activity diagram adalah representasi grafis dari alur kerja tahapan aktifitas. Diagram ini mendukung pilihan tindakan, iterasi dan *concurrency*. Pada pemodelan UML, *activity diagram* dapat digunakan untuk menjelaskan bisnis dan alur kerja operasional secara *step-by-step* dari komponen suatu sistem. *Activity diagram* menunjukkan keseluruhan dari aliran kontrol.

2.12.4 *Class Diagram*

Class diagram merupakan diagram yang selalu ada di permodelan sistem berorientasi objek. *Class* diagram menunjukkan hubungan antar class dalam sistem yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai suatu tujuan.